

```

10  DIM T$(20),S$(40),C(3,2) : P=0 / P = Page counter.
20  / C8$ = Page-seperator string for your printer.
30  C8$=CHR$(12) / Form feed.
40  / C9$ = Page-seperator string for your display screen.
50  CLOSE
55  C9$=CHR$(12) + CHR$(7) : PRINT C9$ / Form feed and beep
60  PRINT" AMSAT ORBITAL PREDICTION PROGRAM de W3IWI - MAY 1980
65  PRINT" Modified for IBM PC by David C. Ensle, KE6ZE - August 1983
70  PRINT" COPYRIGHT 1980 by Dr. Thomas A. Clark, W3IWI"
75  PRINT"          6388 Guilford Road"
80  PRINT"          Clarksville, MD 21029"
85  PRINT" Permission granted for non-commercial use providing"
90  PRINT" credit is given to the author, AMSAT and ORBIT Magazine."
100 PRINT
110 / FNT Function - Returns 2 character printable string corresponding to D.
115 DEF FNT$(D) = CHR$(48 + INT(D/10)) + CHR$(48 + D - 10 * INT(D/10))
120 / FNI Function - Rounds to nearest integer, accounting for sign.
125 DEF FNI(D)=SGN(D) * INT (ABS(D) + .5)
130 / Set starting Day/Time, duration & step size and initialize.
140 INPUT "Start: Year = ",Y : Y=Y/100 : Y=INT(100*(Y-INT(Y)) + .1)
150 IF Y/4=INT(Y/4) THEN LF=1 ELSE LF=0 / LF is leap year flag.
160 INPUT "Month(1-12) = ",M : INPUT "      Day = ",D
165 PRINT : MON=M : DAY=D / Save month and day for later.
170 T$=FNT$(Y) + "/" + FNT$(M) + "/" + FNT$(D) + " AT " : / For printing.
180 / Calculate D8=Day#, Note: month 13=Jan of following year.
190 RESTORE 930 : FOR I=1 TO M : READ D9 : NEXT : D8=D+D9
200 IF M>2 THEN D8=D8+LF : PRINT " Day # is ";D8 : PRINT
210 INPUT"Start: UTC Hours = ",H
220 INPUT"      Min. = ",M : T7=D8 + H/24 + M/1440
230 T$=T$ + FNT$(H) + FNT$(M) + ":00" / For printing.
240 INPUT"Duration: Hours = ",H1
250 INPUT"      Min. = ",M1 : T8=T7 + H1/24 + M1/1440
260 INPUT"Time Step: Min. = ",M2 : T9=M2/1440
270 PRINT "Run will be from day ";T7;" to day ";T8 : PRINT
271 INPUT "Send output to: 'S'creen, 'P'rinter, 'D'isk? = ",OD$
272 IF OD$="S" OR OD$="s" THEN 276
273 IF OD$="P" OR OD$="p" THEN 277
274 IF OD$="D" OR OD$="d" GOTO 280
275 GOTO 271
276 OPEN "SCRN:" FOR OUTPUT AS #1 : C9$=C8$ : P9=1 : GOTO 290
277 OPEN "LPT1:" FOR OUTPUT AS #2 : C9$=C8$ : P9=2 : GOTO 290
280 INPUT "Name for disk file? = ",OD$
281 IF OD$ = "" THEN 271
282 OPEN OD$ FOR OUTPUT AS #3
283 C9$=C8$ : P9=3
290 RESTORE 950 : READ P1,C,R0,F,G0,G1 / Get physical constants.
300 P2=2*(P1) : P0=(P1)/180 : F=1/F
310 READ Y1,G2 : IF Y1=Y THEN 330 ELSE IF Y1>0 THEN 310
320 PRINT "UNABLE TO FIND YEAR ",Y," IN SIDEREAL TIME TABLE" : STOP
330 RESTORE 1090 : READ L9,W9,H9,C$,E8 /Get station data.
340 GOSUB 10520 : GOSUB 860 / Call to FND inits observer's vector.
350 /*****Select parameters for satellite of interest.*****/
360 PRINT"SATELLITE SELECTION MENU" : I=0 : RESTORE 1120 : PRINT
370 READ S$ : IF S$="END" THEN 400 ELSE READ I$ : I=I+1
380 PRINT "Entry #";I;" for ";S$ : PRINT "      ID = ";I$
390 READ D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D : GOTO 370 / D's are dummies.
400 INPUT "Select entry no. ",J : RESTORE 1120 / Get the satellite data.
410 IF (J<1) OR (J>I) THEN 400 / Do a little range checkins.
420 FOR I=1 TO J :READ S$,I$,Y3,D3,H3,M3,S3
430 READ K0,M0,N0,A0,I0,E0,W0,D0,F1 : NEXT
440 PRINT : PRINT "Doppler calculated for freq = ";F1;" Mhz"

```



```

450 INPUT " Change frequency to (0 for default) ",D
460 IF D<>0 THEN F1=D
470 IF Y3=Y THEN 490 ELSE PRINT "ELEMENTS NOT FOR CURRENT YEAR" : STOP
480 ' If D3 is an Integer then epoch is assumed to be in HH,MM,SS format.
485 ' If D3 is not an integer, then D3 is an epoch in days + fractions.
490 PRINT#P9, : PRINT#P9,"Run will be from day ";T7;" to day ";T8
495 PRINT#P9,"Day ";T7;" is day ";DAY;" of month ";MON;" UTC"
500 IF D3=INT(D3) THEN T0=D3 + H3/24 + M3/1440 + S3/86400 ELSE T0=D3
510 PRINT#P9, : PRINT#P9,"Orbital Elements for ";S$ : ' Print out summary.
520 PRINT#P9,"      Reference ID= ";I$ : PRINT#P9,
530 PRINT#P9,"Reference Epoch = ";Y3;"+";T0
540 PRINT#P9,"Starting Epoch = ";Y;"+";T7;"= ";T$
550 PRINT#P9,
560 PRINT#P9,"Parameter";TAB(20);"Reference";TAB(40);"Starting"
570 T=T7 : GOSUB 10110 : GOSUB 10310 ' Initialize FNC & FNM.
580 PRINT#P9,"Orbit Number ",TAB(20);K0;TAB(40);K
590 PRINT#P9,"Mean Anomaly ",TAB(20);M0;TAB(40);M/P0
600 PRINT#P9,"Inclination ",TAB(20);I0
610 PRINT#P9,"Eccentricity ",TAB(20);E0
620 PRINT#P9,"Mean Motion ",TAB(20);N0
630 PRINT#P9,"S.M.A.,km ",TAB(20);A0
640 PRINT#P9,"Arg. Perisee ",TAB(20);W0;TAB(40);W
650 PRINT#P9,"R. A. A. N. ",TAB(20);O0;TAB(40);O
660 PRINT#P9,"Freq.,Mhz ",TAB(20);F1 : K9=9E9 : K8=9E9
670 ' Here follows the actual computation loop.
680 FOR T=T7 TO T8 STEP T9 : K7=INT(T) : GOSUB 10310
690 ' New orbit? If so update the a priori values for W & O for new orbit.
700 IF K=K9 THEN 720
701 GOSUB 10110 : K8=9E9 : K9=9E9
710 ' Above elevation Limit? New date?
720 GOSUB 10370 : GOSUB 10630 ' Init FNK & FNX
725 IF E9<E8 THEN 830
726 IF K7=K8 AND K9=K THEN 770
730 IF K=K9 THEN 760 ELSE GOSUB 860 : K9=K
740 PRINT#P9," U.T.C. AZ EL DOPPLER RANGE HEIGHT LAT LONG PHASE"
750 PRINT#P9,"HH:MM:SS des des Hz km km des des <256>"
760 PRINT#P9,TAB(10),"--- DAY #";K7;" --- ORBIT #";K;" ---"
770 K8=K7 : T4=T-K7 : S4=INT(T4*86400 + .5) : H4=INT(S4/3600 + 1E-6)
780 M4=INT((S4 - H4*3600) / 60 + 1E-6) : S4=S4 - 3600*M4 - 60*M4
790 T$=FNT$(H4) + ":" + FNT$(M4) + ":" + FNT$(S4) ' Printable time strings.
800 F9=-F1*1E6 * R8/C ' F9=Doppler(Hz)=Frequency * Velocity/C
810 U1=FNI(A9) : U2=FNI(E9) : U3=FNI(F9) : U4=FNI(R5) : U5=FNI(R-R0)
815 U6=FNI(L5) : U7=FNI(W5)
820 PRINT#P9,T$;TAB(9);U1;TAB(15);U2;TAB(22);U3;TAB(29);U4;
825 PRINT#P9,TAB(39);U5;TAB(47);U6;TAB(52);U7;TAB(59);M9
830 NEXT : CLOSE : PRINT : INPUT "CONTINUE ? Y/N ",D$
835 IF D$="N" OR D$="n" THEN STOP ELSE GOTO 50
840 '*****
850 ' Page header subroutine.
860 F=F+1
870 PRINT#P9,C9$;C$;" Lat.=";L9;" W.Long.=";W9;" Ht.=";H9;" Ps. #";P
880 PRINT#P9,TAB(10),"-- Minimum Elevation = ";E8;" Des. --"
890 PRINT#P9, : RETURN
900 '*****
910 ' Numerical and Physical Constants.
920 ' Day # at start of each month.
930 DATA 0,31,59,90,120,151,181,212,243,273,304,334,365
940 ' Pi, Velocity of Light, Earth Radius, 1/Earth Flattening Coefficient
950 DATA 3.1415926535 , 2.997925E5 , 6378.160 , 298.25
960 ' GM of Earth in (Orbits/day)^2/km^3 and sidreal/Solar Time rate ratio.

```



```

980 / Greenwich Sidereal Times on Jan 0.0 in days for 1979 thru 1985.
990 DATA 79 , 0.2751843198
1000 DATA 80 , 0.2745212008
1010 DATA 81 , 0.2765959911
1020 DATA 82 , 0.2759328721
1030 DATA 83 , 0.2752697531
1040 DATA 84 , 0.2746066342
1050 DATA 85 , 0.2766814244
1060 DATA 0,0 : ' Terminates Sidereal Table.
1070 / *****
1080 / Station Data: Lat(des),W.Long(des),Height(Meters),Call,Min,Elev.(des)
1085 / DATA 39.185 , 76.936 , 150 , "TEST" , -3
1090 DATA 37.4 , 121.85 , 62 , "KE6ZE" , -1
1100 / *****
1110 / Satellite data tables.
1120 DATA "Phase-3 Transfer Orbit"
1130 DATA "Initial pre-launch guess"
1140 DATA 80 , 144 , 12 , 15 , 00 : ' Epoch (Y,Day#,H,M,S)
1150 DATA 0 : ' Orbit # at Epoch
1160 DATA 0 : ' Mean Anomaly at epoch
1170 DATA 0 , 24364 : ' Mean motion OR semi-major axis
1180 DATA 17.5 : ' Inclination
1190 DATA 0.73 : ' Eccentricity
1200 DATA 189.7 : ' Argument of Perisee
1210 DATA 212.35 : ' R.A.A.N.
1220 DATA +145.81 : ' Frequency for doppler calculation
1230 / *****
1240 DATA "OSCAR-10" : ' Satellite
1250 DATA "DC-10" : ' Parameter set ID
1260 DATA 83 , 215 , 00 , 00 , 00 : ' Epoch (Y,Day#,H,M,S)
1270 DATA 0 : ' Orbit # at Epoch
1280 DATA 331.5371 : ' Mean Anomaly at epoch
1290 DATA 2.05848540 , 0 : ' Mean motion OR semi-major axis
1300 DATA 25.6674 : ' Inclination
1310 DATA .6046100 : ' Eccentricity
1320 DATA 191.1818 : ' Argument of Perisee
1330 DATA 248.6091 : ' R.A.A.N.
1340 DATA -289.2 : ' Frequency for doppler calculation
1350 / *****
1360 DATA "OSCAR-10"
1370 DATA "AMSAT net 21 AUG 83, bad RAAN"
1380 DATA 83 , 223.48180313,0,0,0 : ' Epoch (Y,Day#,H,M,S)
1390 DATA 121 : ' Orbit # at Epoch
1400 DATA 137.3886 : ' Mean Anomaly at epoch
1410 DATA 2.05851948 , 0 : ' Mean motion OR semi-major axis
1420 DATA 26.2024 : ' Inclination
1430 DATA 0.6039871 : ' Eccentricity
1440 DATA 193.5343 : ' Argument of Perisee
1450 DATA 24.8822 : ' R.A.A.N.
1460 DATA -289.2 : ' Frequency for doppler calculation
1470 / *****
1480 DATA "Oscar-10" : ' Satellite
1490 DATA "MH 8-4-83" : ' Parameter set ID
1500 DATA 83,216.49,0,0,0 : ' Epoch (Y,Day#,H,M,S)
1510 DATA 0 : ' Orbit # at Epoch
1520 DATA 355.8520 : ' Mean Anomaly at epoch
1530 DATA 2.05881910,0 : ' Mean motion OR semi-major axis
1540 DATA 26.1720 : ' Inclination
1550 DATA .6036399 : ' Eccentricity
1560 DATA 191.6770 : ' Argument of Perisee

```



```

1580 DATA -289.2 ; ' Frequency for doppler calculation
1590 *****
9000 DATA "END" : ' End of the satellite data table.
9010 *****
10000 ' Orbit Determination and Utility Routines.
10010 ' FCA Subroutine.
10020 ' Calculates INVERSE TANGENT in proper quadrant, just like
10030 ' in the FORTRAN ATAN2 function.
10040 IF XX<0 THEN 10070
10050 IF XX>0 THEN 10080
10060 ' The two cases where XX=0.
10061 IF YY>=0 THEN FCA=(P1)/2 ELSE FCA=3*(P1)/2
10062 RETURN
10070 ' Cases lying in quadrant 2&3.
10071 FCA = (P1) + ATN(YY/XX)
10072 RETURN
10080 ' Cases lying in quadrant 1&4
10081 IF YY>=0 THEN FCA = ATN(YY/XX) ELSE FCA = P2 + ATN(YY/XX)
10082 RETURN
10100 *****
10110 ' FNC Subroutine
10120 ' Routine to initialize the C(J,K) coordinate rotation matrix
10130 ' and other parameters associated with the orbital elements.
10140 IF NO > 0.1 THEN A0=((G0 / (NO^2)) ^ (1/3)) ' Given NO=Mean Motion.
10150 IF NO <= 0.1 THEN NO=SQR(G0 / (A0^3)) ' Given A0=Semi-Major axis.
10160 E2=1-E0^2 : E1=SQR(E2) : Q0=M0/360 + K0 ' Q0=Initial orbit phase.
10170 ' Account for nodal effects due to lumpy gravity field due to the
10180 ' flattened, oblate spheroidal, figure of the earth.
10190 K2=9.95 * ((R0/A0)^3.5) / (E2^2)
10200 ' Update elements to current epoch & evaluate their SIN/COS's.
10210 S1=SIN(I0*P0) : C1=COS(I0*P0) ' I0=Inclination (deg).
10220 D=D0 - (T-T0) * K2 * C1
10230 S0=SIN(D*P0) : C0=COS(D*P0) ' D=R.A.A.N. (deg)
10240 W=W0 + (T-T0) * K2 * (2.5 * (C1^2) - .5)
10250 S2=SIN(W*P0) : C2=COS(W*P0) ' W=Argument of Perisee (deg).
10260 ' Set up coordinate rotation matrix for the current orbit.
10270 C(1,1)= + (C2*C0) - (S2*S0*C1) : C(1,2)= - (S2*C0) - (C2*S0*C1)
10280 C(2,1)= + (C2*S0) + (S2*C0*C1) : C(2,2)= - (S2*S0) + (C2*C0*C1)
10290 C(3,1)= + (S2*S1) : C(3,2)= + (C2*S1) : RETURN
10300 *****
10310 ' FNM Subroutine.
10320 ' Function to evaluate M=MEAN ANOMALY in (0-2PI) range,
10330 ' K=Perisee Passage Counter (=Orbit ##) & M9=Modulo 256
10340 ' orbital phase compatible with General Beacon Telemetry.
10350 Q=N0*(T-T0)+Q0 : K=INT(Q) : M9=INT((Q-K)*256) : M=(Q-K)*P2 : RETURN
10360 *****
10370 ' FNK Subroutine.
10380 ' Routine to solve KEPLER's equation given M & return
10390 ' the Satellite's GEOCENTRIC coordinates.
10400 E=M + E0*SIN(M) + .5 * (E0^2) * SIN(2*M) ' Initial trial value.
10410 ' Iteration loop to solve transcendental Kepler's equation.
10420 S3=SIN(E) : C3=COS(E) : R3=1 - E0*C3 : M1=E - E0*S3
10430 M5=M1-M : IF ABS(M5) < 1E-6 THEN 10450 ELSE E=E - M5/R3 : GOTO 10420
10440 ' Now set satellite's XYZ coordinates.
10450 X0=A0*(C3-E0) : Y0=A0*E1*S3 : R=A0*R3 ' In the plane of the orbit.
10460 ' Rotate from Orbit plane to INERTIAL CELESTIAL Coordinates.
10470 X1=X0*C(1,1)+Y0*C(1,2) : Y1=X0*C(2,1)+Y0*C(2,2) : Z1=X0*C(3,1)+Y0*C(3,2)
10480 ' Rotate thru current GHA of ARIES, convert to GEOCENTRIC coordinates.
10490 G7=T*G1 + G2 : G7=(G7-INT(G7)) * P2 : S7=-SIN(G7) : C7=COS(G7)
10500 X= + (X1*C7) - (Y1*S7) : Y= + (X1*S7) + (Y1*C7) : Z=Z1 : RETURN

```

```

1280 DATA -289.0
1290 *****
1300 DATA "END" : End of the satellite data table.
1310 *****
1320 : Orbit Determination and Utility Routines.
1330 : FCA Subroutine.
1340 : Calculated INVERSE TANGENT in proper quadrant. Just like
1350 : in the FORTRAN ATAN2 Function.
1360 IF XX=0 THEN 10070
1370 IF XX=0 THEN 10080
1380 : The two cases where XX=0.
1390 IF YY=0 THEN FCA=(PI)/2 ELSE FCA=3*(PI)/2
1400 RETURN
1410 : Cases lying in quadrant 2&3.
1420 FCA = (PI) + ATN(YY/XX)
1430 RETURN
1440 : Cases lying in quadrant 1&4.
1450 IF YY=0 THEN FCA = ATN(YY/XX) ELSE FCA = PI + ATN(YY/XX)
1460 RETURN
1470 *****
1480 : FNC Subroutine.
1490 : Routine to initialize the C(3,K) coordinate rotation matrix
1500 : and other parameters associated with the orbital elements.
1510 IF NO > 0.1 THEN AO=((60 / (NO*2)) * (1/3)) : Given NO=Mean Motion.
1520 IF NO < 0 : Error. NO must be positive.
1530 E2=1-E02 : Eccentricity squared.
1540 : Account for node longitude.
1550 : Flattened.
1560 E2=9.95 * ((R/AO)^(2.5)) / (E02)
1570 D=0.0 - (T-T0) * K2 * O1
1580 S0=SIN(O*PI) : C0=COS(O*PI) : D=R.A.M. (deg)
1590 W=W0 + (T-T0) * K2 * (C1/2) - (E)
1600 E2=SIN(W*PI) : C2=COS(W*PI) : W=Argument of Perigee (deg).
1610 : Set up coordinate rotation matrix for the current orbit.
1620 C(1,1)=+ (C2*K0) + (E2*K0) : C(1,2)=+ (E2*K0) - (E2*K0*O1)
1630 C(2,1)=+ (E2*K0) + (E2*K0*O1) : C(2,2)=+ (E2*K0) - (E2*K0*O1)
1640 C(3,1)=+ (E2*K0) : C(3,2)=+ (E2*K0) : RETURN
1650 *****
1660 : FNM Subroutine.
1670 : Function to evaluate MEAN ANGLE in (0-2PI) range.
1680 : K=Perigee Passage Counter (=Orbit #) : M=Modulo 256
1690 : orbital phase compatible with General Bessel Tables.
1700 D=NO*(T-T0) : K=INT(D) : M=INT((D-K*256) / M) : M=0-K*256 : RETURN
1710 *****
1720 : FNM Subroutine.
1730 : Routine to solve KEPLER'S equation given M & return
1740 : the satellite's GEOCENTRIC coordinates.
1750 E=M + E0*SIN(M) + .5 * (E0^2) * SIN(2*M) : Initial trial value.
1760 : Iteration loop to solve transcendental Kepler's equation.
1770 E2=SIN(E) : E3=COS(E) : R3=1 - E0*K2 : M1=E - E0*E2
1780 M2=M1 - (E3*(R3)) : IF ABS(M2) < 1E-6 THEN 10450 ELSE E=E2 : GOTO 10430
1790 : Now set satellite's XYZ coordinates.
1800 X0=X0+E3 : Y0=Y0-E3 : Z0=Z0+E3
1810 : Rotate from orbit plane to inertial frame.
1820 X1=X0*(1+Y0*(1/2)) : Y1=X0*(2/3)+Y0*(2/3) : Z1=X0*(2/3)+Y0*(2/3)
1830 : Rotate thru current GHA of ARIES, convert to GEOCENTRIC coordinates.
1840 D2=K01 + G2 : G2=(G2-INT(G2)) * PI : E2=SIN(G2) : C2=COS(G2)

```

Digitized by the Internet Archive

in 2025 with funding from

Amateur Radio Digital Communications, Grant 151

<https://archive.org/details/amsat-orbit-prog>

```

10520 / FNO Subroutine.
10530 / Routine to evaluate OBSERVER's GEOCENTRIC coordinates,
10540 / where X-axis=GREENWICH, Y-axis=VU-land & Z-axis=North pole.
10550 L8=L9*P0 : S9=SIN(L8) : C9=COS(L8) / Initial GEOCENTRIC coordinates.
10560 S8=SIN(-W9*P0) : C8=COS(W9*P0) / W9=West Longitude.
10570 / Followins accounts for flattened oblate spheroid earth.
10580 R9=R0*(1 - (F/2) + (F/2) * COS(2*L8)) + H9/1000 / H9=height(meters)
10590 / Followins makes L8 be the GEOCENTRIC Latitude.
10600 L8=ATN((1-F)^2 * S9/C9) : Z9=R9 * SIN(L8)
10610 X9=R9 * COS(L8) * C8 : Y9=R9 * COS(L8) * S8 : RETURN
10620 / *****
10630 / FNX Subroutine
10640 / Routine to extract all the parameters you might ever need.
10650 / First set vector from observer to satellite.
10660 X5= (X-X9) : Y5= (Y-Y9) : Z5= (Z-Z9) : R5=SQR(X5^2 + Y5^2 + Z5^2)
10670 / Finite difference the range (R5) to get the velocity (R8).
10680 IF T6<>T THEN R8=((R6-R5)/(T6-T)) / 86400 ELSE R8=-9E9
10690 R6=R5 : T6=T / Save current range & time for next time thru.
10700 / Now rotate into OBSERVER's LOCAL coordinates
10710 / where X8=North, Y8=East, Z8=Up (Left-Handed system).
10720 Z8= + (X5*C8*C9) + (Y5*S8*C9) + (Z5*S9)
10730 X8= - (X5*C8*S9) - (Y5*S8*S9) + (Z5*C9) : Y8= + (Y5*C8) - (X5*S8)
10740 S5=Z8/R5 : C5=SQR(1-S5*S5) : E9=ATN(S5/C5) / P0 / E9=Elevation.
10750 XX=X8 : YY=Y8 : GOSUB 10010
10751 A9=FCA / P0 / FCA resolves quadrant for A9=Azimuth.
10760 XX=X : YY=Y : GOSUB 10010
10761 W5=360-FCA / P0 / W5=Satellite Sub Point (SSP) W. Longitude.
10770 B5=Z/R : L5=ATN(B5 / (SQR(1 - B5^2))) / P0 / SSP Latitude.
10780 RETURN / Note R-R0=Satellite's altitude above mean spheroid.
10790 / *****
10800 / FNT$ Moved to beginning of the program.
10810 / FNI Moved to beginning of the program.
19999 END

```


From fortune!amd70!decwrl!decvax!senrad!srkermit!masscomp!c!yde!ihnp4!houxm!mhux

Subject: Satellite Orbital Elements

Newsgroups: net.ham-radio

Satellite: oscar-10

Catalog number: 14129

Epoch time: 83285.50000000

Wed Oct 12 12:00:00.000 1983 UTC

Element set: MH 10-11-83

Inclination: 26.0050 deg

RA of node: 236.6270 deg

Eccentricity: 0.6049226

Arg of perisee: 210.5700 deg

Mean anomaly: 17.2220 deg

Mean motion: 2.05854010 rev/day

Decay rate: 0 rev/day²

Epoch rev: 249

Semi major axis: 26105.707 km

Anom period: 699.524872 min

Aposee: 35520.558 km

Perisee: 3936.693 km

Translate freq: 581.0047 mhz

Invert: 1

Beacon: 145.8100 mhz

Satellite: oscar-9

Catalog number: 12888

Epoch time: 83287.08040297

Fri Oct 14 01:55:46.817 1983 UTC

Element set: 523

Inclination: 97.5569 deg

RA of node: 253.8344 deg

Eccentricity: 0.0001137

Arg of perisee: 236.6515 deg

Mean anomaly: 123.4644 deg

Mean motion: 15.23344898 rev/day

Decay rate: 6.486e-05 rev/day²

Epoch rev: 11189

Semi major axis: 6870.864 km

Anom period: 94.528823 min

Aposee: 508.165 km

Perisee: 506.602 km

Beacon: 145.8250 mhz

Satellite: oscar-8

Catalog number: 10703

Epoch time: 83284.83733089

Tue Oct 11 20:05:45.389 1983 UTC

Element set: 805

Inclination: 98.7580 deg

RA of node: 291.9257 deg

Eccentricity: 0.0008120

Arg of perisee: 7.0984 deg

Mean anomaly: 353.0321 deg

Mean motion: 13.96578109 rev/day

Decay rate: -1.7e-07 rev/day²

Epoch rev: 28552

Semi major axis: 7280.998 km

Anom period: 103.109163 min

Aposee: 909.084 km

Perisee: 897.260 km

Translate freq: 581.0974 mhz

Invert: 1
Beacon: 435.0965 mhz

Satellite: rs-5
Catalog number: 12999
Epoch time: 83283.14307672
Mon Oct 10 03:26:01.829 1983 UTC
Element set: 123
Inclination: 82.9551 deg
RA of node: 279.6636 deg
Eccentricity: 0.0009189
Arg of perisee: 257.4527 deg
Mean anomaly: 102.5470 deg
Mean motion: 12.05052030 rev/day
Decay rate: 4e-08 rev/day²
Epoch rev: 7970
Semi major axis: 8033.846 km
Anom period: 119.496915 min
Aposee: 1683.153 km
Perisee: 1668.388 km

Satellite: rs-6
Catalog number: 13002
Epoch time: 83281.06115721
Sat Oct 8 01:28:03.983 1983 UTC
Element set: 65
Inclination: 82.9585 deg
RA of node: 274.8561 deg
Eccentricity: 0.0049822
Arg of perisee: 172.9858 deg
Mean anomaly: 187.1915 deg
Mean motion: 12.13556218 rev/day
Decay rate: 4e-08 rev/day²
Epoch rev: 8001
Semi major axis: 7996.245 km
Anom period: 118.659521 min
Aposee: 1658.253 km
Perisee: 1578.575 km

Satellite: rs-7
Catalog number: 13001
Epoch time: 83274.03818428
Sat Oct 1 00:54:59.122 1983 UTC
Element set: 124
Inclination: 82.9582 deg
RA of node: 282.0638 deg
Eccentricity: 0.0021349
Arg of perisee: 204.5528 deg
Mean anomaly: 155.4546 deg
Mean motion: 12.08678651 rev/day
Decay rate: 4e-08 rev/day²
Epoch rev: 7884
Semi major axis: 8017.757 km
Anom period: 119.138366 min
Aposee: 1660.366 km
Perisee: 1626.132 km

Satellite: rs-8
Catalog number: 12998
Epoch time: 83286.04966178
Thu Oct 13 01:11:30.778 1983 UTC

Element set: 247
 Inclination: 82.9552 deg
 RA of node: 279.4784 deg
 Eccentricity: 0.0017295
 Ars of perisee: 301.8725 deg
 Mean anomaly: 58.0684 deg
 Mean motion: 12.02940035 rev/day
 Decay rate: 4e-08 rev/day²
 Epoch rev: 7991
 Semi major axis: 8043.253 km
 Anom period: 119.706715 min
 Aposee: 1694.209 km
 Perisee: 1666.388 km

Satellite: rs-3
 Catalos number: 12997
 Epoch time: 83282.43160566
 Sun Oct 9 10:21:30.729 1983 UTC
 Element set: 80
 Inclination: 82.9556 deg
 RA of node: 272.6532 deg
 Eccentricity: 0.0059122
 Ars of perisee: 153.1700 deg
 Mean anomaly: 207.2443 deg
 Mean motion: 12.15580726 rev/day
 Decay rate: 4e-08 rev/day²
 Epoch rev: 8032
 Semi major axis: 7987.359 km
 Anom period: 118.461898 min
 Aposee: 1660.728 km
 Perisee: 1566.282 km

Satellite: rs-4
 Catalos number: 13000
 Epoch time: 83283.16787147
 Mon Oct 10 04:01:44.950 1983 UTC
 Element set: 145
 Inclination: 82.9669 deg
 RA of node: 278.6280 deg
 Eccentricity: 0.0016879
 Ars of perisee: 224.7168 deg
 Mean anomaly: 135.2533 deg
 Mean motion: 12.06666291 rev/day
 Decay rate: 4e-08 rev/day²
 Epoch rev: 7981
 Semi major axis: 8026.674 km
 Anom period: 119.337054 min
 Aposee: 1672.505 km
 Perisee: 1645.409 km

Satellite: noaa-7
 Catalos number: 12553
 Epoch time: 83270.81463539
 Tue Sep 27 19:33:04.498 1983 UTC
 Element set: 223
 Inclination: 99.0087 deg
 RA of node: 234.9254 deg
 Eccentricity: 0.0013923
 Ars of perisee: 115.4993 deg
 Mean anomaly: 244.7631 deg
 Mean motion: 14.12927711 rev/day

Decay rate: 1.44e-06 rev/day²
Epoch rev: 11666
Semi major axis: 7224.689 km
Anom period: 101.916042 min
Aposee: 873.598 km
Perisee: 853.480 km
Beacon: 137.6200 mhz

Satellite: noaa-8
Catalog number: 13923
Epoch time: 83281.58488859
Sat Oct 8 14:02:14.374 1983 UTC
Element set: 57
Inclination: 98.7385 deg
RA of node: 310.3123 deg
Eccentricity: 0.0017861
Arg of perisee: 65.2796 deg
Mean anomaly: 295.0227 deg
Mean motion: 14.22340763 rev/day
Decay rate: 1.21e-06 rev/day²
Epoch rev: 2756
Semi major axis: 7192.741 km
Anom period: 101.241562 min
Aposee: 844.681 km
Perisee: 818.987 km
Beacon: 137.5000 mhz


```
program orbit(input,output);
```

```
const
```

```
    bisnumber = 32700;
```

```
type
```

```
    string2 = array [1..2] of char;
```

```
    string6 = array [1..6] of char;
```

```
    string7 = array [1..7] of char;
```

```
    string8 = array [1..8] of char;
```

```
var
```

```
    s1, s2, x, y, x1, y1, z1, xi : real;
```

```
    meanmotion, meanmajoraxis, meanperiodic : real;
```

```
    orbitinc, eccentricity, argsofperisee : real;
```

```
    refraan, referoch, twoepi, eccanomaly : real;
```

```
    radstodes, t7, t8, t9, xsat, ysat : real;
```

```
    clisht, s0, e1, e2, a0, m, k2, s0, s1, s2 : real;
```

```
    r0, c0, c1, c2, destorads, w, o, epoch, a, f, x8, y8 : real;
```

```
    c3, s3, e, m5, z, r, r3, s, s7, s7, c7, z8 : real;
```

```
    e2, l8, r8, s8, c9, h9, l9, r9, s9, w9, u9, y9, z9 : real;
```

```
    e1, e9, e9, b5, c5, l5, r5, r6, s5, t6, a5, x5, y5, z5 : real;
```

```
    doeshift, d, d9, d8, e8, f1 : real;
```

```
    flas, yr, mo, min, min1, da, leapflas, h, h1 : integer;
```

```
    m2, k, k0, k7, k8, k9, m9, y3, d3, b3, u3, number, p : integer;
```

```
    ystr, mstr, dstr : array [1..2] of char;
```

```
    collision : array [1..6] of char;
```

```
    epochs : array [1..7] of char;
```

```
    datestr : array [1..8] of char;
```

```
    c : array [1..3,1..2] of real;
```

```
procedure credits;
```

```
{ give credits, set slow, etc. }
```

```
begin
```

```
    writeln;
```

```
    writeln(' Satellite Orbital Prediction Program');
```

```
    writeln(' written by:');
```

```
    writeln(' Lyle Johnson, WA7GXD, June, 1983');
```

```
    writeln(' from a BASIC program by:');
```

```
    writeln(' Tom Clark, W3IWI, May, 1980');
```

```
    writeln(' and published by AMSAT in ORBIT No. 6');
```

```
    writeln(' Permission granted for non-commercial use providing');
```

```
    writeln(' credit is given to the author, AMSAT and ORBIT magazine.');
```

```
    writeln
```

```
end { credits }
```

```
function invtan (denom, num : real) : real;
```

```
{ calculate inverse tangent function in proper quadrant }
```

```
begin
```

```
    if denom<0
```

```
    then
```



```

      invtan := pi + arctan(num/denom)
    else
      if denom=0
      then
        if num>=0
        then
          invtan := pi/2
        else
          invtan := 3*pi/2
        else
          if num>=0
          then
            invtan := arctan(num/denom)
          else
            invtan := two*pi + arctan(num/denom)
        end { invtan };
  end { invtan };

  procedure initrotationmatrix;

  < routine to initialize the the c[[1]] coordinate rotation matrix
    and other parameters associated with the orbital elements >

  begin
    if meanmotion>0.1
    then
      semimajoraxis := (exp((1/3)*ln(a0/exp(meanmotion))))
    else
      meanmotion := sqrt(a0/exp(3*ln(semimajoraxis)));
    e2 := 1-sqr(eccentricity);
    e1 := sqrt(e2);
    a0 := semimajoraxis/360 + k0;

    < account for nodal effects due to lunar gravitational field due to
      the flattened, oblate spheroid figure of the earth >

    k2 := 9.95*(exp(3.5*ln(a0/semimajoraxis)))/sqr(e2);

    < update elements to to current epoch and evaluate their SIN and COS >

    s1 := sin(orbitline*destorads);
    c1 := cos(orbitline*destorads);
    o := refran - (epoch-refepoch) * k2 * c1;
    s0 := sin(o*destorads);
    c0 := cos(o*destorads);
    w := ergofperisee + (epoch-refepoch) * k2 * (2.5 * cos(c1) - 0.5);
    s2 := sin(w*destorads);
    c2 := cos(w*destorads);

    < set up coordinate rotation matrix for the current orbit >

    c[1,1] := (c2*c0) - (s2*s0*c1);
    c[1,2] := -(s2*c0) - (c2*s0*c1);
    c[2,1] := (c2*s0) + (s2*c0*c1);
    c[2,2] := -(s2*s0) + (c2*c0*c1);
    c[3,1] := (s2*s1);
    c[3,2] := (c2*s1);

  end { initrotationmatrix };

```



```
procedure setmeananomaly;
```

```
{ evaluates meananomaly in 0 to two*pi range,
  k is perisee passage counter (orbit #),
  m9 is modulo 256 orbital phase compatible with phase3 telemetry }
```

```
begin
```

```
  a := meanmotion*(epoch-refepoch) + a0;
  k := trunc(a);
  m9 := trunc((a-k)*256);
  m := (a-k) * two*pi
```

```
end { setmeananomaly };
```

```
procedure kepler;
```

```
{ routine to solve Kepler's equation, given m, and return
  the satellite's GEOCENTRIC coordinates }
```

```
begin
```

```
  { initial guess }
```

```
  e := m + eccentricity*sin(m) + 0.5 * sin(eccentricity) * sin(2*m);
```

```
  { iterative loop to solve transcendental Kepler's equation }
```

```
  m5 := 0 { set up for first pass through loop };
```

```
  repeat
```

```
    s3 := sin(e);
    c3 := cos(e);
    r3 := 1-eccentricity*c3;
    m1 := e-eccentricity*s3;
    m5 := m1-m;
    e := e - m5/r3;
```

```
  until abs(m5)<1e-04;
```

```
  { now set satellite's xyz coordinates in orbital plane }
```

```
  xset := semimajoraxis * (c3-eccentricity);
  yset := semimajoraxis * e1 * s3;
  r := semimajoraxis * r3;
```

```
  { rotate from orbital plane to INERTIAL CELESTIAL coordinates }
```

```
  x1 := xset * cC1,12 + yset * cC1,21;
  y1 := xset * cC2,12 + yset * cC2,21;
  z1 := xset * cC3,12 + yset * cC3,21;
```

```
  { rotate through GHA of Aries, convert to GEOCENTRIC coordinates }
```

```
  s7 := epoch* s1 + s2;
  s7 := (s7 - trunc(s7)) * two*pi;
  s7 := -sin(s7);
  c7 := cos(s7);
  x := (x1*c7) - (y1*s7);
  y := (x1*s7) + (y1*c7);
  z := z1;
```

```
end { kepler };
```



```
procedure observer;
```

```
{ calculate OBSERVER'S GEOCENTRIC coordinates,
  where Xaxis = GREENWICH, Yaxis = VU-LAND and
  Zaxis = NORTH POLE. }
```

```
begin
```

```
l8 := l9*destorads;
s9 := sin(l8);
c9 := cos(l8);
s8 := sin(-w9*destorads);
c8 := cos(w9*destorads);
r9 := r0*(1-(f/2)+(f/2)*cos(2*l8))+h9/1000;
l8 := arctan(s9*(1-f)*s9/c9);
z9 := r9*sin(l8);
x9 := r9*cos(l8)*c8;
y9 := r9*cos(l8)*s8;
```

```
end { observer };
```

```
procedure setparameters;
```

```
{ routine to extract all the parameters you might ever need }
```

```
begin
```

```
{ first set the vector from the observer to the satellite }
```

```
x5 := v-x9;
y5 := y-y9;
z5 := z-z9;
r5 := sqrt(sqr(x5) + sqr(y5) + sqr(z5));
```

```
{ finite difference the range, r5, to get the velocity, r8 }
```

```
if epoch<t6
then
  r8 := ((r6-r5)/(t6-epoch))/(184400.0);
else
  r8 := -r9;
```

```
{ save current range and time for next time through }
```

```
r6 := r5;
t6 := epoch;
```

```
{ now rotate into OBSERVER'S LOCAL coordinates: where
  X8 = north, Y8 = east and Z8 = up [left-handed system] }
```

```
z8 := (x5*c8*c9) + (y5*s8*c9) + (z5*s9);
x8 := -(x5*c8*s9) - (y5*s8*s9) + (z5*c9);
y8 := (y5*c8) - (x5*s8);
s5 := z8/r5;
c5 := sqrt(1-s5*s5);
e9 := arctan(s5/c5)*radstodeg;
w9 := invtan(x8,y8)*radstodeg;
w5 := 360 - invtan(x,y)*radstodeg;
b5 := z/r;
l5 := arctan(b5/(sqrt(1-sqr(b5))))*radstodeg;
```


end { setparameters };

procedure dayorbit;

{ write out a banner if day and/or orbit number changes }

begin

writeln(' - - - Day Nr ',k7:3,' - - - Orbit ',k,' - - -');

end { dayorbit };

procedure paramheader;

{ Print the column headings for desired data }

begin

writeln;

writeln(' U.T.C. AZ EL Doppler Range Height Est Lng Phase');

writeln('HHMM:SS degrees Hz km km degrees (255)');

dayorbit

end { paramheader };

procedure orbitsummary;

{ Print out data about satellite and selected epoch }

begin

writeln('Reference Epoch = ',y3:2,' + ',refepoch:14:10);

writeln('Starting Epoch = ',yr:2,' + ',t7:14:10);

writeln;

writeln('Parameter: Reference' :23: 'Starting' :20);

epoch := t7;

initrotationmatrix;

setmeananomaly;

writeln('Orbit Number',k0:20;k:20);

writeln('Mean Anomaly',meananomaly:20:10;#redstodes:20:10);

writeln('Inclination',orbitinc:21:2);

writeln('Eccentricity',eccentricity:20:10);

writeln('Mean Motion',meanmotion:21:10);

writeln('S.M.A., km',semimajoraxis:22:1);

writeln('Ars. Perisee',arsofperisee:20:10;w:20:10);

writeln('E, A, A, N.',refraa:21:10;w:20:10);

writeln('Free, MHz',f1:23:3)

end;

procedure inttostr(n : integer; var s : string2);

{ converts integer to a two-place string }

begin

s[2] := chr(n mod 10 + ord('0'));

n := n div 10;

s[1] := chr(n mod 10 + ord('0'));

end { inttostr };


```
procedure stationParams;
```

```
  { set default station parameters }
```

```
begin
```

```
  l9 := 39.185 { latitude in degrees, + = north, - = south };
```

```
  w9 := 76.936 { west longitude in degrees };
```

```
  h9 := 150 { height MSL in meters };
```

```
  e8 := -3 { minimum satellite elevation to print };
```

```
  callsign := 'W3IWI ';
```

```
  l9 := 37.5 { latitude in degrees, + = north, - = south };
```

```
  w9 := 122.4 { west longitude in degrees };
```

```
  h9 := 10 { height MSL in meters };
```

```
  e8 := 0 { minimum satellite elevation to print };
```

```
  callsign := 'KA6M ';
```

```
end;
```

```
procedure newPage;
```

```
  { print new page header with page number }
```

```
begin;
```

```
  writeln;
```

```
  write(callsign, ' Lat. = ', l9:6:3, ' W. Long. = ', w9:6:3);
```

```
  write(' Ht. = ', h9:5:1);
```

```
  p := p+1;
```

```
  writeln(' Page #', p:3);
```

```
  writeln(' - - - Minimum Elevation = ', e8:4:2, ' Deg. - - -');
```

```
  writeln
```

```
end { newPage };
```

```
procedure timestring (time : real; var result : string7 );
```

```
  { convert current epoch to printable string }
```

```
var
```

```
  timeint : integer;
```

```
  numsecs, nummins, numhrs : integer;
```

```
  fracday, tempsecs : real;
```

```
  secstr, minstr, hrstr : array [1..2] of char;
```

```
begin
```

```
  timeint := trunc(time);
```

```
  fracday := time-timeint;
```

```
  tempsecs := fracday*86400.0;
```

```
  numhrs := trunc(tempsecs/3600+1e-6);
```

```
  nummins := trunc((tempsecs-numhrs*3600.0)/60 + 1e-6);
```

```
  numsecs := trunc(tempsecs-3600.0*numhrs-60*nummins);
```

```
  { now convert integers to strings... }
```

```
  inttostr(numhrs, hrstr);
```

```
  inttostr(nummins, minstr);
```

```
  inttostr(numsecs, secstr);
```



```

/ ...and concatenate to a single 7 place string

```

```

result[1] := hrstr[1];
result[2] := hrstr[2];
result[3] := minstr[1];
result[4] := minstr[2];
result[5] := ':';
result[6] := secstr[1];
result[7] := secstr[2];

```

```

end ( timestring );

```

```

procedure ac10transfer;

```

```

( initialize satellite data for Phase 3 transfer orbit )

```

```

begin

```

```

w3 := 83;
d3 := 170;
h3 := 12;
m3 := 00;
s3 := 00;
k0 := 6;
meananomaly := 313.5510;
meanmotion := 0;
semimajoraxis := 24253.781;
orbitinc := 2.5590;
eccentricity := 0.7288272;
argsofperisee := 180.7290;
refraan := 246.1490;
f1 := 145.61;
flag := 0;

```

```

end ( ac10transfer );

```

```

procedure ac8;

```

```

( initialize satellite data for Oscar 8 )

```

```

begin

```

```

w3 := 83;
d3 := 133;
h3 := 12;
m3 := 37;
s3 := 48;
k0 := 26440;
meananomaly := 296.6894;
meanmotion := 0;
semimajoraxis := 7283.299;
orbitinc := 92.7578;
eccentricity := 0.0007984;
argsofperisee := 63.8803;
refraan := 147.8904;
f1 := 435.0964;
flag := 0;

```

```

end ( ac8 );

```



```
procedure Uo9;
```

```
< initialize satellite data UoSat Oscar 9 >
```

```
begin
```

```

  u3 := 83;
  d3 := 131;
  h3 := 09;
  m3 := 23;
  s3 := 13;
  k0 := 3820;
  meananomaly := 282.0193;
  meanmotion := 0;
  semimajoraxis := 4530.058;
  orbitinc := 97.3426;
  eccentricity := 0.0004573;
  arsofperisee := 73.1334;
  refrasn := 97.4153;
  fl := 145.8250;
  flag := 0;

```

```
end < Uo9 >;
```

```
procedure rs5;
```

```
< initialize satellite data for RadioSport 5 >
```

```
begin
```

```

  u3 := 83;
  d3 := 120;
  h3 := 04;
  m3 := 01;
  s3 := 06;
  k0 := 6007;
  meananomaly := 126.5394;
  meanmotion := 0;
  semimajoraxis := 8037.799;
  orbitinc := 82.9681;
  eccentricity := 0.0009302;
  arsofperisee := 233.4807;
  refrasn := 8.2360;
  fl := 0;
  flag := 0;

```

```
end < rs5 >;
```

```
procedure rs6;
```

```
< initialize satellite data for RadioSport 6 >
```

```
begin
```

```

  u3 := 83;
  d3 := 123;
  h3 := 14;
  m3 := 20;
  s3 := 02;
  k0 := 6097;
  meananomaly := 119.2001;

```



```

meanmotion := 0;
semimajoraxis := 8000.184;
orbitinc := 82.9637;
eccentricity := 0.0050379;
argofperisee := 141.2644;
refraan := 1.8637;
f1 := 0;
flag := 0;

```

```
end { rs6 };
```

```
procedure rs7;
```

```
{ initialize satellite data for RadioSport 7 }
```

```
begin
```

```

g3 := 83;
d3 := 131;
h3 := 0;
m3 := 00;
s3 := 35;
k0 := 6156;
meananomaly := 229.4669;
meanmotion := 0;
semimajoraxis := 8021.717;
orbitinc := 82.9629;
eccentricity := 0.0022005;
argofperisee := 139.8076;
refraan := 0.3421;
f1 := 0;
flag := 0;

```

```
end { rs7 };
```

```
procedure rs8;
```

```
{ initialize satellite data for RadioSport 8 }
```

```
begin
```

```

g3 := 83;
d3 := 128;
h3 := 0;
m3 := 40;
s3 := 47;
k0 := 6091;
meananomaly := 96.8142;
meanmotion := 0;
semimajoraxis := 8047.201;
orbitinc := 82.9612;
eccentricity := 0.0018664;
argofperisee := 263.0991;
refraan := 3.6189;
f1 := 0;
flag := 0;

```

```
end { rs8 };
```

```
procedure satdata;
```


{ select satellite and initialize parameters for it }

begin

repeat

readln(number);

writeln('Satellite selection number was ',number;2);

flag := -1;

case number of

1 : ao10transfer;

2 : up9;

3 : ao8;

4 : rs5;

5 : rs6;

6 : rs7;

7 : rs8;

end (case);

if flag <> 0

then

begin

writeln;

writeln('*** --> Invalid selection! <-- ***');

writeln

end;

until flag=0;

end { satdata };

procedure checkdoppler;

{ this procedure verifies the spot frequency to be displayed for
doppler offset }

begin

writeln;

writeln('Doppler calculated for freq = ',f1;7;3,' MHz,');

readln(d);

if d <> 0

then

f1 := d;

writeln('Change frequency to: ',f1;7;3,' MHz,');

end { checkdoppler };

procedure physconstants;

{ initialize physical constants }

begin

pi := 4*arctan(1);

centuradi := pi/180;

radetodeg := 1/centuradi;

twopi := 2*pi;


```

c1ight := 2.997923e3;
r0 := 6379,140;
f := 298,25;
f := 1/f;
s0 := 7.5369793e13;
s1 := 1.00273790941;
s2 := 0.2745212008

```

```

end { physconstants };

```

```

procedure setuser;

```

```

{ get session-specific data from user }

```

```

begin

```

```

  repeat

```

```

    readln(y);
    x := y/100;
    yr := trunc(100*(y-trunc(y)) + 0.1);
    writeln('Input start year as two digits (ex: 83) --> ');
    if yr/4 = trunc(yr/4.0)
    then
      leapflag := 1
    else
      leapflag := 0;
    flag := 1;

```

```

  case x of

```

```

    79 : begin
      s2 := 0.2751843198;
      flag := 0;
    end;
    80 : begin
      s2 := 0.2745212008;
      flag := 0;
    end;
    81 : begin
      s1 := 0.2765959911;
      flag := 0;
    end;
    82 : begin
      s2 := 0.2759328721;
      flag := 0;
    end;
    83 : begin
      s2 := 0.2752697531;
      flag := 0;
    end;
    84 : begin
      s2 := 0.2746066342;
      flag := 0;
    end;
    85 : begin
      s2 := 0.2766314744;
      flag := 0;
    end;
  end;
end;

```



```

if flag=0
then
begin
writein
write(*** --: Unable to find year 19712/1
writein --: Elapsed Time (sec): ***);
writein
end;

```

until flag=1

repeat

```

readin(20);
writein(10); Start month as per above table and --- 19712/1

```

flag is 1;

case no of

```

1 1 begin
    ad is 0;
    flag is 0
end;

```

```

2 1 begin
    ad is 3;
    flag is 0
end;

```

```

3 1 begin
    ad is 29;
    flag is 0
end;

```

```

4 1 begin
    ad is 4;
    flag is 0
end;

```

```

5 1 begin
    ad is 120;
    flag is 0
end;

```

```

6 1 begin
    ad is 171;
    flag is 0
end;

```

```

7 1 begin
    ad is 181;
    flag is 0
end;

```

```

8 1 begin
    ad is 71;
    flag is 0
end;

```

```

9 1 begin
    ad is 14;
    flag is 0
end;

```

```

10 1 begin
    ad is 272;
    flag is 0
end;

```

```

11 1 begin
    ad is 94;

```



```

        flag := 0
    end;
12 : begin
    mo := 1999;
    day := 0;
    end;
13 : begin
    mo := 1999;
    day := 0;
    end;
end : loop 24;

if flag = 0
then
begin
    writeln;
    writeln('*** --> Invalid month detected! <-- ***');
    writeln;
end;
until flag=0;

readln(da);
write ('Input start day as two digits (eg, 23) --> ',da);
inttostr(mo,mstr);
inttostr(da,dstr);
datestr(11,1,1999,11);
datestr(12,1,1999,12);
datestr(13,1,1999,1);
datestr(14,1,1999,2);
datestr(15,1,1999,3);
datestr(16,1,1999,4);
datestr(17,1,1999,5);
datestr(18,1,1999,6);
datestr(19,1,1999,7);
datestr(20,1,1999,8);
datestr(21,1,1999,9);
datestr(22,1,1999,10);
datestr(23,1,1999,11);
datestr(24,1,1999,12);

{ now calculate Julian date }

d8 := d8 + da;
if mo>2
then
begin
    d8 := d8 + (mo-1)*31;
    write('Day Nr = ',d8:4:4);
    readln;
    write ('Start: UTC Hours = ',h:2);
    readln(min);
    writeln('Min. = ',min:2);
    t7 := d8 + h/24 + min/1440;
    inttostr(t7,mstr);
    inttostr(min,dstr);
    datestr := concat(datestr,mstr,dstr,':00');
    readln;
    write('Duration: Hours = ',h1:3);
    readln(min1);
    writeln('Min. = ',min1:2);
    t8 := t7 + h1/24 + min1/1440;
    readln(m2);
    write ('Time step: Min. = ',m2:3);
    t9 := m2/1440;
    writeln('From ',t7:7:4,' to ',t8:7:4);

```



```
end { setuser};
```

```
begin { orbit }
```

```
  p := 0;
```

```
  credits;
```

```
  physconstants;
```

```
  stationparams;
```

```
  observer;
```

```
  satdata;
```

```
  setuser;
```

```
  while wr<>w3 do
```

```
    begin
```

```
      writeln;
```

```
      writeln('*** --> Orbit elements not from current year! <-- ***');
```

```
      writeln;
```

```
      setuser;
```

```
      stationparams;
```

```
      observer;
```

```
      satdata
```

```
    end;
```

```
  checkdoppler;
```

```
  referepoch := d3+(h3/24)+(m3/1440)+(s3/(86400.0));
```

```
  orbitsummary;
```

```
  k9 := bisnumber;
```

```
  k8 := bisnumber;
```

```
{ offset epoch for first pass through loop }
```

```
epoch := epoch-t9;
```

```
{ now do computation/print loop for desired period }
```

```
repeat
```

```
  epoch := epoch+t9;
```

```
  k7 := trunc(epoch);
```

```
  setmeananomaly;
```

```
  if k<>k9
```

```
    then begin
```

```
      initrotationmatrix;
```

```
      k8 := bisnumber;
```

```
      k9 := bisnumber
```

```
    end;
```

```
  kepler;
```

```
  satparameters;
```

```
  if e8<e9
```

```
  then begin
```

```
    if not ((k7=k8) and (k9=k))
```

```
    then begin
```

```
      if k<>k9
```

```
        then begin
```

```
          newpage;
```

```
          paramheader;
```

```
          k9 := k
```



```
        end
        else
            daworbit
        end;
    k8 := k7;
    dorshift := -f1*1e6*r8/clisht;
    timestrins(epoch, epochs);
    write(epochs,round(a9):6,round(e9):4,dorshift:9:1,r5:9:1);
    writeln((r-r0):9:1,round(l5):6,round(w5):4,m9:5)
end

until epoch>=t8

end.
```

